

KAKAS BANTU PENDETEKSI KESALAHAN TANDA BACA PADA KARYA TULIS ILMIAH

Ratih Nur Esti Anggraini¹⁾, Mohammad Ahmaluddin Zinni²⁾, dan Siti Rochimah³⁾

Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember

Jl. Teknik Kimia, Gedung Teknik Informatika,

Kampus ITS Sukolilo, Surabaya, 60111

e-mail: ratih_nea@if.its.ac.id¹⁾, siti@if.its.ac.id³⁾

ABSTRAK

Jumlah karya ilmiah yang dihasilkan oleh akademisi dan peneliti di Indonesia semakin banyak, terutama setelah diterbitkannya surat edaran Dirjen DIKTI tahun 2012 dimana karya ilmiah dijadikan sebagai syarat kelulusan mahasiswa S1, S2 dan S3. Namun demikian, tidak semua karya ilmiah tersebut memiliki kualitas yang baik. Masih banyak karya ilmiah yang belum memenuhi standar baku Ejaan Yang Disempurnakan (EYD). Pada artikel ini, penulis mengembangkan sebuah kakas bantu untuk mendeteksi kesalahan tanda baca pada karya ilmiah, khususnya yang berbahasa Indonesia, sesuai dengan EYD. Aplikasi dirancang agar dapat mendeteksi kesalahan tanda baca pada tulisan karya ilmiah dengan format .doc atau .docx serta dapat menghasilkan keluaran berupa arsip Microsoft Word dengan tambahan hasil telaah pemeriksaan tanda baca yang dibangkitkan secara otomatis. Deteksi kesalahan tanda baca menggunakan metode pencarian kata dengan algoritma Boyer-Moore. Aplikasi kakas bantu telah diuji coba dengan hasil rata-rata nilai presisi sistem sebesar 0,6806, recall sebesar 0,969 dan akurasi sistem sebesar 0,9636. Hasil tersebut menunjukkan bahwa aplikasi sudah mampu mendeteksi adanya kesalahan tanda baca meskipun masih ada keterbatasan deteksi karena tidak semua aturan tanda baca dicakup dalam pemeriksaannya.

Kata Kunci: Boyer-Moore, Dokumen Microsoft Word, Karya Ilmiah, Tanda Baca.

ABSTRACT

The number of scientific papers produced by academics and researchers in Indonesia is increasing, especially after the Director General of Higher Education letter in 2012 where the scientific work used as a graduation requirement for undergraduate, master and doctoral degrees. However, not all of these scientific works that have a good quality. There are still many scientific works that do not meet the standards of Spelling Enhanced. In this article, we developed a tool to detect punctuation errors in the scientific work, especially Indonesian article, according to the Spelling Enhanced. This tool was designed to detect errors of punctuation in writing scientific work with .doc or .docx format and can produce output in the form of Microsoft Word files with additional punctuation error detection that was automatically generated. Boyer-Moore algorithm is used to detect punctuation errors. This tool has been tested with an average precision of 0.6806, the recall of 0.969 and accuracy 0.9636. These results indicate that this tool is able to detect any punctuation errors although there are limitations detection because not all of the rules of punctuation are used in the detection.

Keywords: Boyer-Moore, Microsoft Word Document, Scientific Works, Punctuation.

I. PENDAHULUAN

Karya ilmiah dalam bentuk makalah telah menjadi syarat kelulusan di perguruan tinggi, sesuai dengan surat edaran Direktorat Jenderal Pendidikan Tinggi 152/E/T/2012 tanggal 27 Januari 2012 tentang publikasi karya ilmiah [1]. Dalam surat edaran tersebut, mahasiswa S1 diharuskan menghasilkan karya ilmiah yang terbit pada jurnal ilmiah, mahasiswa S2 harus menghasilkan karya ilmiah untuk jurnal nasional, sedangkan mahasiswa S3 harus menghasilkan karya ilmiah yang terbit di jurnal internasional sebagai syarat kelulusan. Diterbitkannya surat edaran tersebut berdampak pada semakin banyaknya karya ilmiah yang dihasilkan oleh para akademisi dari berbagai bidang ilmu dan lingkup penelitian [2].

Namun demikian, tidak semua karya ilmiah yang dihasilkan tersebut memiliki kualitas yang baik. Masih ada pula beberapa karya ilmiah yang salah dalam penulisan kata, ejaan serta aturan atau kaidah yang baku sesuai dengan Ejaan Yang Disempurnakan (EYD).

Saat ini, aplikasi koreksi kesalahan berbahasa pada karya ilmiah berbahasa Inggris sudah ada dalam pengolahan kata seperti Microsoft Word. Aplikasi ini dapat menunjukkan kesalahan gramatikal dengan garis bawah berwarna hijau dan kesalahan pengejaan dengan garis bawah berwarna merah. Namun, untuk karya ilmiah berbahasa Indonesia belum tersedia aplikasi koreksi kesalahan berbahasa seperti yang sudah ada untuk karya ilmiah berbahasa Inggris. Padahal karya ilmiah yang baik harus memenuhi kaidah atau standar baku seperti yang telah ditentukan berdasarkan EYD.

Kesalahan berbahasa Indonesia yang sering terjadi pada umumnya adalah kesalahan tanda baca. Entah kesalahan yang dibuat karena tidak mengetahui aturan EYD yang baku tentang penggunaan tanda baca, atau karena memang

lupa dalam melakukan koreksi kesalahan tanda baca pada tulisan yang dibuat. Padahal kesalahan tanda baca ini kemungkinan dapat menimbulkan ambiguitas bagi sang pembaca, atau bahkan dapat pula mengubah arti suatu kalimat atau pernyataan.

Artikel ini akan membahas suatu aplikasi koreksi kesalahan penggunaan tanda baca dalam karya ilmiah berbahasa Indonesia agar karya ilmiah yang dihasilkan memenuhi kaidah ragam tulis baku yang baik dan benar. Ketersediaan aplikasi koreksi ini diharapkan dapat membantu dunia keilmiah Indonesia dalam upaya meningkatkan kualitas tulisan karya ilmiah dosen, mahasiswa, dan masyarakat Indonesia pada umumnya.

II. TINJAUAN PUSTAKA

A. Tanda Baca

Tanda baca adalah simbol yang berperan untuk menunjukkan struktur dan organisasi suatu tulisan, dan juga intonasi serta jeda yang dapat diamati sewaktu pembacaan.

Beberapa jenis tanda baca yang penting dan sering muncul pada karya ilmiah antara lain adalah sebagai berikut.

- Titik (.) berfungsi untuk menandai akhir kalimat berita, atau untuk keperluan singkatan, gelar, dan angka-angka.
- Koma (,) berfungsi untuk memisahkan anak kalimat atau hal-hal yang disebutkan dalam kalimat, juga untuk keperluan singkatan, gelar, dan angka-angka.
- Tanda tanya (?) dipakai pada akhir kalimat tanya.
- Tanda hubung(...-...) berfungsi untuk menghubungkan penggalan kata, kata ulang atau rentang suatu nilai.
- Tanda titik koma (;) dapat dipakai untuk memisahkan bagian-bagian kalimat yang sejenis dan setara.
- Titik dua (:) berfungsi untuk mengawali penguraian suatu kalimat [3].

B. Algoritma Boyer-Moore

Algoritma pencarian kata Boyer-Moore dikembangkan oleh Bob Boyer dan J Strother Moore pada tahun 1977. Algoritma ini akan bertambah cepat jika kata yang dicari panjang. Efisiensi yang algoritma ini punya berasal dari fakta bahwa, setiap pencocokan yang gagal antara teks dan kata yang dicari, algoritma ini menggunakan informasi yang didapat dari proses awal untuk melewati karakter-karakter yang tidak cocok.

Algoritma Boyer-Moore didasarkan pada dua heuristik pencarian yaitu heuristik *looking-glass* dan heuristik *character-jump*. Heuristik *looking-glass* melakukan perbandingan suatu karakter akhir pada kata *w* dengan suatu karakter pada teks *s*. Jika karakter tersebut sama maka jendela karakter akan berjalan mundur pada kedua string dan memeriksa kembali kedua karakter. Sedangkan heuristik *character-jump* melakukan suatu aksi ketika terdapat perbandingan antara dua karakter yang berbeda.

Ada dua aksi yang tergantung pada teks *s* dan kata *w* yang dimiliki. Jika *p* merupakan karakter pada *s* yang sedang diproses yang tidak cocok, maka ada dua kemungkinan aksi. Pertama jika *p* terdapat dalam kata *w*, geser kata *w* agar karakter *p* pada *w* (pilih yang paling akhir jika ada lebih dari satu) sejajar dengan karakter *p* pada *s*. Kedua jika *p* tidak terdapat dalam kata *w*, geser kata *w* sehingga karakter awal kata *w* sejajar dengan karakter setelah *p* pada *s* [4].

C. Primary Interop Assembly References

Primary Interop Assembly Reference adalah pustaka .NET yang dapat digunakan untuk melakukan operasi-operasi yang ada pada aplikasi Microsoft Word dengan menggunakan bahasa C#. Operasi-operasi yang dapat dilakukan dengan Microsoft Office Word Primary Interop Reference ini beberapa di antaranya adalah membuat arsip baru, membuat salinan dari satu arsip Microsoft Word ataupun melakukan penghapusan karakter [5].

D. DocX

DocX adalah pustaka .NET yang memungkinkan pengembang untuk memanipulasi dokumen Word 2007/2010/2013, dengan cara yang lebih mudah dan intuitif. Docx mempunyai performa yang cepat dan ringan. DocX juga tidak memerlukan Microsoft Office Word yang terpasang sebagai prasyarat.

Beberapa fungsi yang ada di dalam DocX ini seperti menambahkan, menghapus atau mengganti kata yang ada di dokumen dan disertai dengan format standart text. Terdapat pula fungsi perubahan paragraf seperti indentasi, format penulisan rata kanan atau rata kiri serta penjajaran (*alignment*) [6].

E. Presisi, Recall dan Akurasi

Untuk dapat mengetahui tingkat performansi suatu sistem yang mampu mendapatkan kembali informasi-informasi tertentu dapat diketahui menggunakan perhitungan presisi dan *recall*. Presisi merupakan probabilitas

informasi yang relevan dari semua informasi yang didapatkan kembali oleh sistem. Rumus untuk menghitung presisi dan *recall* dapat dilihat pada Persamaan (1) dan Persamaan (2) sedangkan hubungannya dengan tabel *contingency* dapat dilihat pada Persamaan (3) dan Persamaan (4).

$$precision = \frac{relevant\ items\ retrieved}{retrieved\ items} \quad (1)$$

$$recall = \frac{relevant\ items\ retrieved}{relevant\ items} \quad (2)$$

Sedangkan *Recall* merupakan probabilitas informasi relevan yang didapatkan kembali oleh sistem dibandingkan jumlah informasi-informasi yang relevan.

Dua persamaan di atas dapat diperjelas dengan melihat tabel *contingency*. Tabel *contingency* Presisi dan *Recall* dapat dilihat pada **Error! Reference source not found.** Kemudian, untuk rumus presisi (P) berdasarkan tabel *contingency* didapat:

$$P = \frac{TP}{TP + FP} \quad (3)$$

Sedangkan untuk rumus *recall* (R) berdasarkan tabel *contingency* didapat:

$$R = \frac{TP}{TP + FN} \quad (4)$$

Alternatif lain yang mungkin dapat diketahui oleh pengguna dalam menilai suatu sistem temu kembali informasi yakni dengan menggunakan akurasi [7]. Rumus perhitungan akurasi berdasarkan tabel *contingency* dapat dilihat pada Persamaan 5.

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

III. ANALISIS DAN PERANCANGAN

A. Deskripsi Umum Sistem

Sistem yang dibuat yaitu berupa aplikasi *desktop* dengan antarmuka Windows Form. Sistem menghasilkan keluaran berupa arsip Microsoft Word dengan tambahan hasil telaah pemeriksaan tanda baca yang dibangkitkan secara otomatis. Masukan program berupa tulisan karya ilmiah dalam bentuk arsip Microsoft Word dengan format .doc atau .docx. Deskripsi sistem ini adalah sebagai berikut.

1. Pengguna memberikan masukan berupa arsip karya tulis ilmiah dalam bentuk dokumen .doc atau .docx.
2. Sistem mengidentifikasi kesalahan tanda baca yang ada pada dokumen.
3. Sistem memberikan tanda latar belakang merah pada posisi kesalahan tanda baca dan memberikan hasil telaah pemeriksaan tanda baca kesalahan fitur Comment Microsoft Word pada paragraf yang memiliki kesalahan tanda baca di dalamnya.
4. Sistem menyimpan keluaran dokumen baru yang berisi karya tulis ilmiah yang sama ditambah dengan tanda latar belakang merah pada posisi kesalahan tanda baca serta memberikan hasil telaah pemeriksaan tanda baca kesalahan fitur Comment Microsoft Word pada paragraf yang memiliki kesalahan tanda baca di dalamnya.

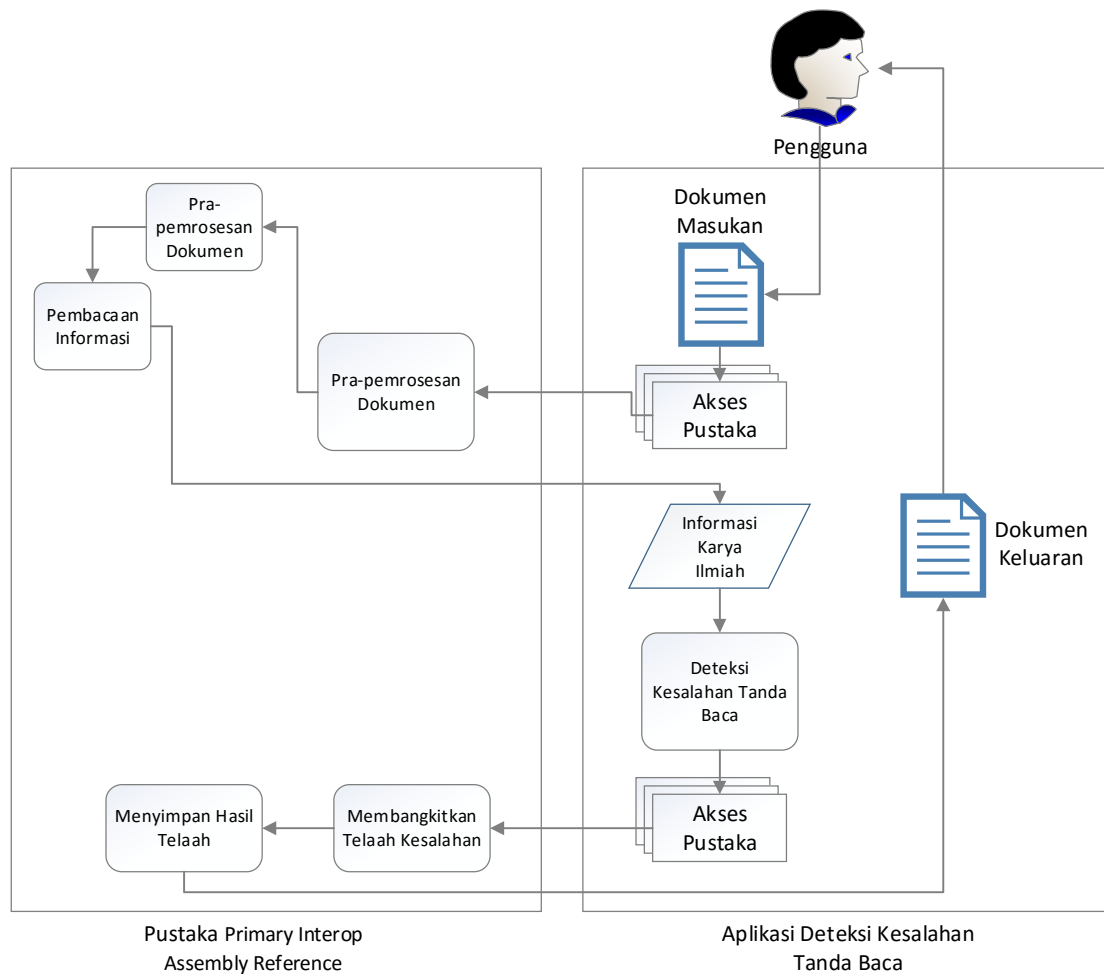
Dengan adanya sistem ini, pengguna dapat terbantu dalam melakukan telaah kesalahan tanda baca pada karya ilmiah sehingga nantinya kualitas pembuatan karya ilmiah semakin membaik. Secara ilustratif, arsitektur sistem dapat dilihat melalui **Error! Reference source not found.**

B. Proses Deteksi Tanda Baca

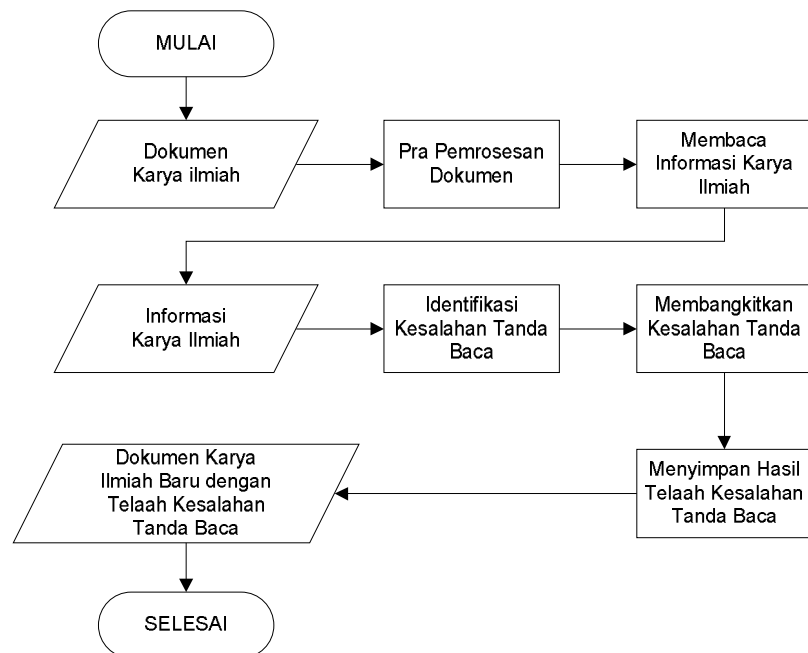
Pada bagian ini akan dijelaskan proses jalannya sistem yang dikembangkan menggunakan bahasa pemrograman C#. Sistem dimulai ketika terdapat masukan berupa dokumen karya ilmiah berekstensi .doc dan .docx. Setelah itu akan dilakukan pembacaan informasi yang didapat dari dokumen tersebut menggunakan Primary Interop Assembly References.

Tabel I. Tabel Contingency Presisi dan Recall

		Real Value	
		Relevant/True	Non-relevant/False
Prediction Value	Relevant/True	True Positive (TP)	False Positive (FP)
	Non-relevant/False	False Negative (FN)	True Negatives (TN)



Gambar 1. Arsitektur Sistem



Gambar 2. Diagram Alir Sistem Deteksi Kesalahan Tanda Baca

Proses berikutnya adalah pencarian kesalahan tanda baca. Proses pencarian sendiri menggunakan algoritma pencarian string Boyer-Moore. Alur jalannya sistem dapat dilihat pada Gambar 2. Penjelasan rinci dari tiap proses adalah sebagai berikut.

1. Pra-pemrosesan Dokumen

Pada tahap ini dokumen masukan dari pengguna akan mengalami pra-pemrosesan. Pra-pemrosesan pada kasus ini adalah proses penghilangan/penghapusan informasi yang tidak relevan atau tidak perlu dilakukan pendeteksian tanda baca. Beberapa informasi yang dihapus yakni: judul; informasi pengarang; abstraksi; tabel; gambar; *caption*; rumus ilmiah; sitasi; dan daftar pustaka. Hasil dari pra-pemrosesan dokumen dapat dilihat dengan tampilan tiap bab.

2. Membaca Informasi Karya Ilmiah

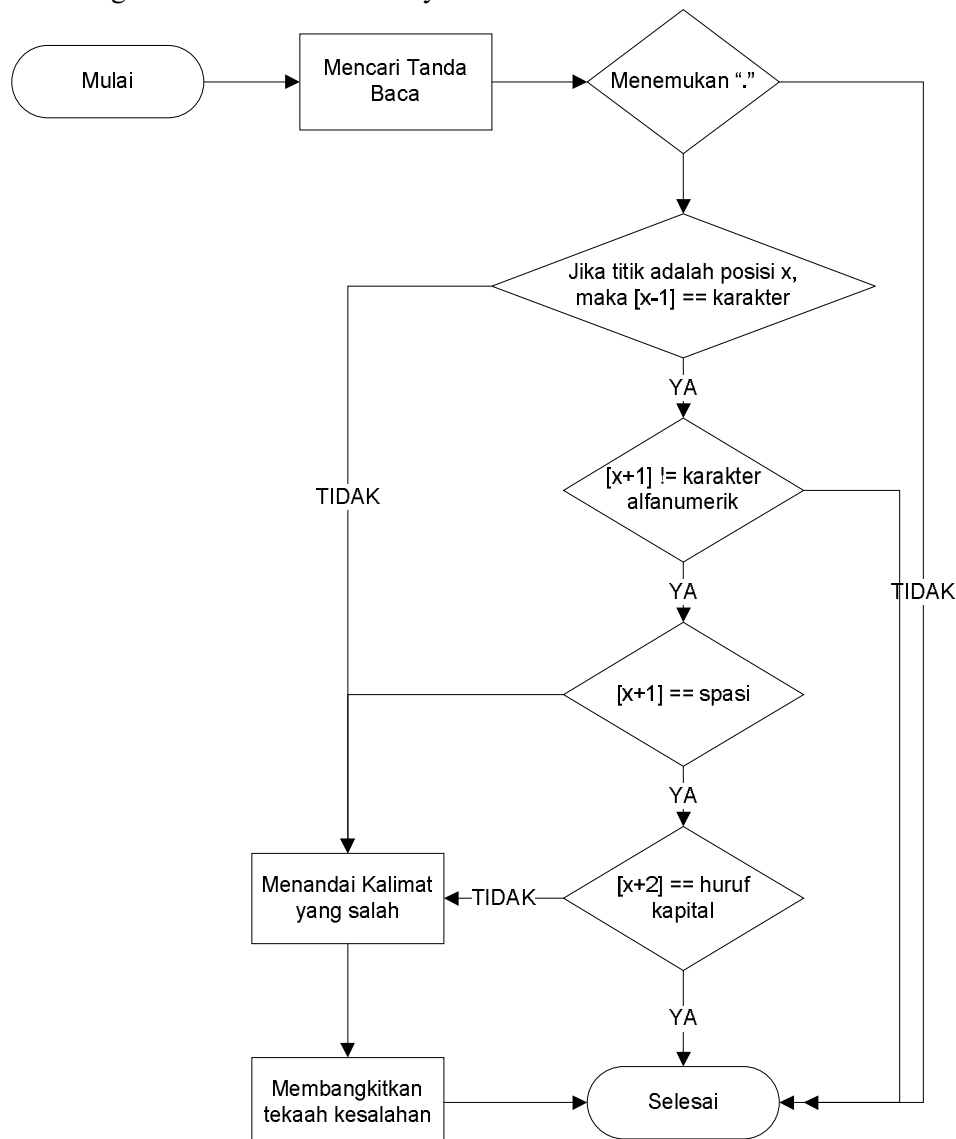
Pada tahap ini, informasi yang ada pada dokumen karya ilmiah akan diambil. Sehingga didapatkan kumpulan kalimat yang dapat diidentifikasi karakternya.

3. Identifikasi Kesalahan Tanda Baca

Dari kumpulan kalimat yang telah didapat dari proses ekstraksi dokumen, akan dicari tanda baca yang ada. Proses pencarian tanda baca menggunakan algoritma pencarian Boyer-Moore. Jika ditemukan tanda baca yang ada pada kumpulan kata, maka kemudian akan diperiksa syarat-syarat penggunaan tanda bacanya. Jika terdapat kesalahan, informasi kesalahan akan dimasukkan dalam daftar kesalahan tanda baca dari dokumen. Proses pemeriksaan berlangsung sampai semua kata telah diperiksa. Berdasarkan definisi penggunaan tanda baca pada karya ilmiah yang telah dijelaskan pada Bab sebelumnya, telah dibuat pula alur jalannya program untuk pemeriksaan tiap tanda baca. Contoh alur pemeriksaan tanda baca titik (.) dapat dilihat pada Gambar 3.

4. Membangkitkan Telaah Kesalahan Tanda Baca

Apabila pada dokumen karya ilmiah yang dideteksi terdapat kesalahan, maka akan terdapat informasi yang dimasukkan pada daftar kesalahan tanda baca. Setelah itu daftar kesalahan tersebut akan dibangkitkan dan disematkan pada dokumen karya ilmiah baru yang isinya sama dengan dokumen masukan awal ditambah dengan hasil telaah yang sesuai dengan kesalahan tanda bacanya.



Gambar 3. Diagram Alir Pemeriksaan Tanda Baca Titik (.)

5. Menyimpan Hasil Telaah Kesalahan

Setelah dokumen karya ilmiah yang disertai telaah kesalahan tanda baca dibuat oleh sistem, pengguna dapat menyimpannya pada tempat penyimpanan lokal komputer. Dokumen disimpan dalam bentuk arsip .docx.

IV. PENGUJIAN

A. Skenario Pengujian

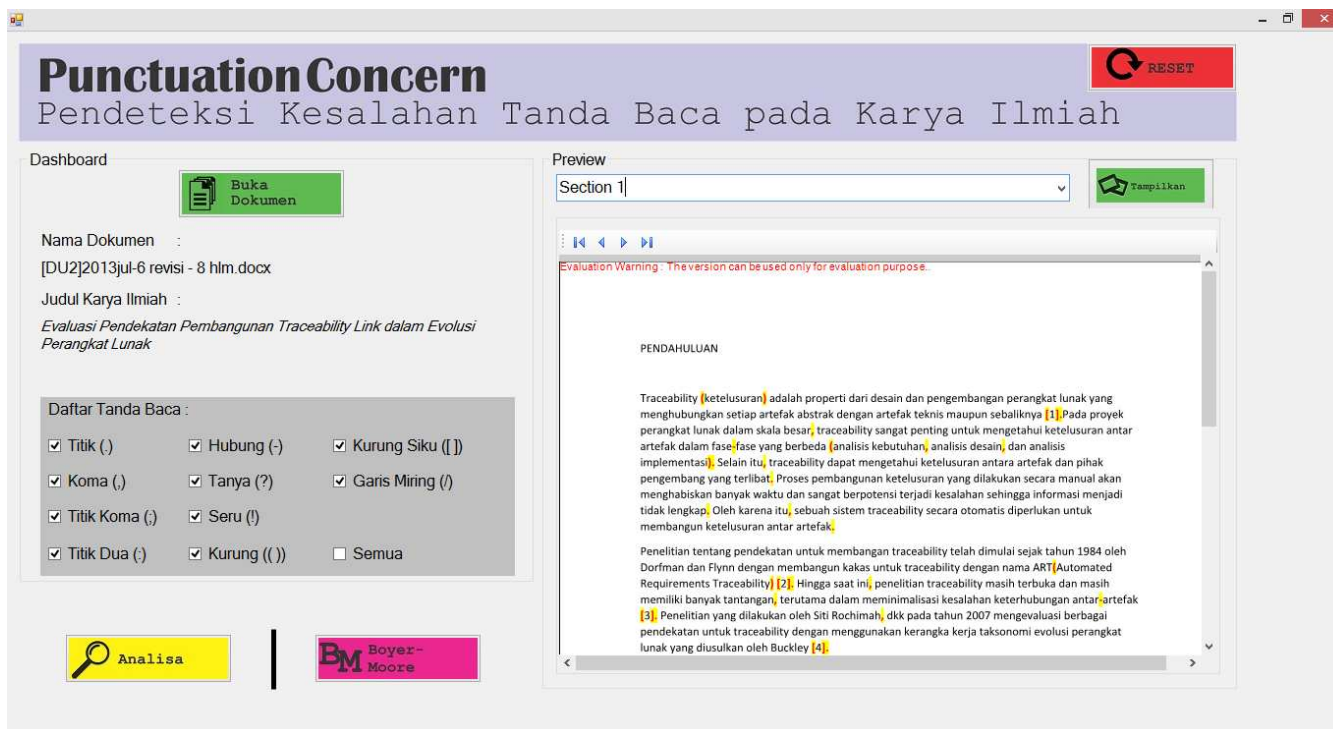
Lingkungan pengujian sistem pada pembuatan aplikasi kakas bantu ini adalah sebagai berikut.

- Prosesor : Intel Core i3 M370 CPU @ 2.40GHz
- Memori : 4.00 GB
- Aplikasi Penunjang: Microsoft Office Word 2013

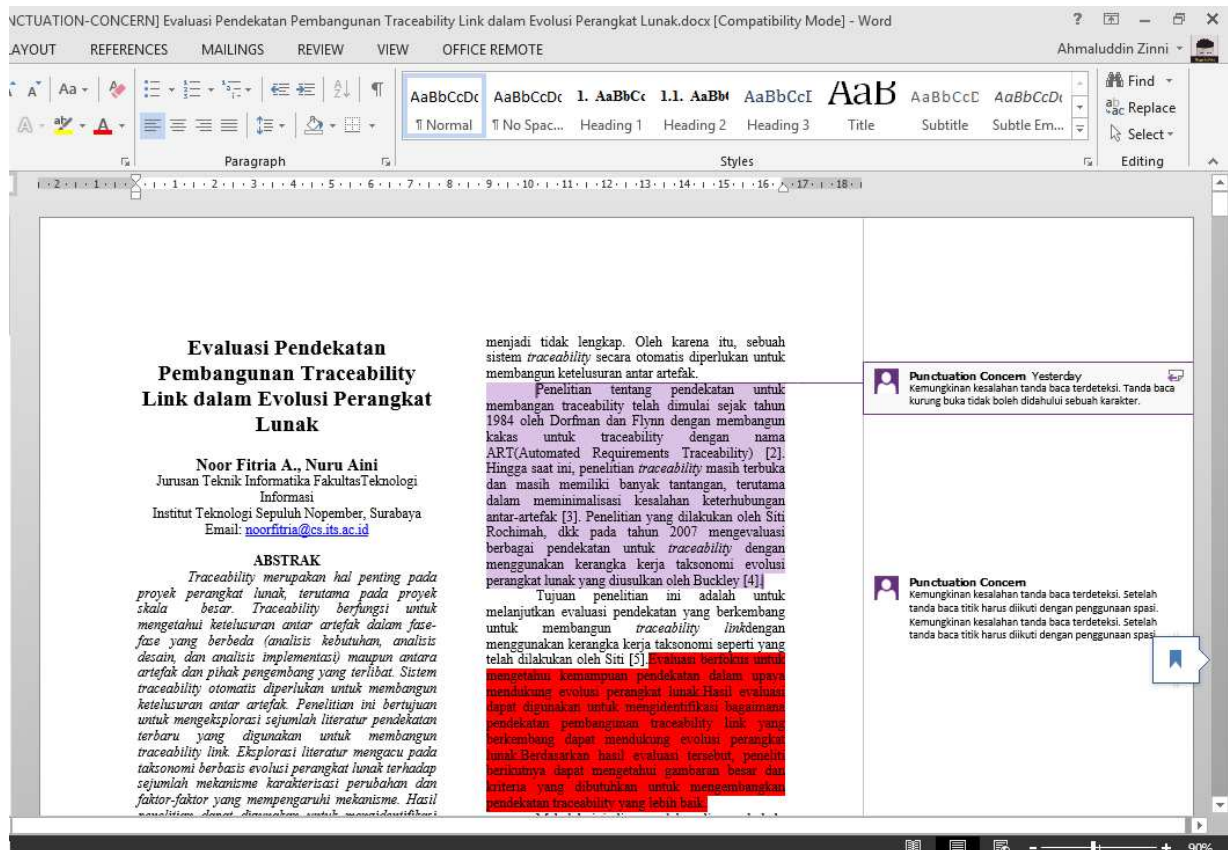
Pengujian pada aplikasi dilakukan dengan pengujian kebutuhan fungsionalitas. Pengujian fungsionalitas menggunakan metode kotak hitam (*black box*) dengan hasil yang didapat seluruh fungsional sistem dapat berjalan sebagaimana mestinya. Gambar 4 menunjukkan aplikasi kakas bantu pendeteksi kesalahan tanda baca setelah dilakukan pra pemrosesan, sedangkan hasil deteksi kesalahan tanda baca ditunjukkan pada Gambar 5.

Pengujian terhadap hasil deteksi juga dilakukan dengan membandingkan hasil deteksi sistem dengan hasil analisa manual yang dilakukan oleh penelaah karya tulis ilmiah (*reviewer*). *Reviewer* membaca terlebih dahulu karya ilmiah yang dijadikan data pengujian. Kemudian *reviewer* mendeteksi kesalahan apa saja yang ada pada karya ilmiah tersebut dengan menuliskan nomor bab, paragraf dan kalimat yang ditengarai mempunyai kesalahan tanda baca. Untuk mengukur performa dari sistem perlu dilakukan perhitungan tingkat presisi, *recall* dan akurasi.

Dari dokumen keluaran sistem yang berupa karya ilmiah dengan tambahan telaah dan tanda latar belakang merah pada kalimat yang terdapat penggunaan tanda baca yang salah, kemudian dibandingkan dengan dokumen hasil telaah secara manual dari ahli. Hasil dari identifikasi kemudian dituliskan dalam laporan hasil uji yang didalamnya terdapat identifikasi keterangan deteksi tanda baca yang bernilai *true positive*, *false positive*, *false negative* dan *true negative*. Dari identifikasi itu kemudian dihitung nilai presisi, *recall* dan akurasi. Pengujian dilakukan terhadap 5 data uji yang diambil secara acak. Detail informasi dan hasil perbandingan data uji 1 dapat dilihat pada Tabel 2 dan Tabel 3. Sedangkan hasil perhitungan rata-rata tingkat presisi, *recall* dan akurasi dari sistem berdasarkan 5 data uji dapat dilihat pada Tabel 4.



Gambar 4. Kakas Bantu Pendeteksi Kesalahan Tanda Baca



Gambar 5. Hasil Luaran dari Kakas Bantu Deteksi Kesalahan Tanda Baca

Tabel II. Informasi Data Uji 1

Nomor Data Uji	: 1
Nama Dokumen Data Uji	: 2014jan-7 revisi - 4 hlm.docx
Judul Karya Ilmiah	: Studi Pustaka untuk Menemukan Hubungan antara Evolusi Perangkat Lunak dengan Framework COBIT 4.1
Jumlah Tanda Baca	: 269 tanda baca

Tabel III. Hasil Perbandingan dan Identifikasi Daftar Kesalahan Tanda Baca pada Data Uji 1

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	22	18
	FALSE	4	225

Tabel IV. Daftar Nilai Presisi, Recall dan Akurasi dari Hasil Pengujian

No.	Data Uji	Presisi	Recall	Akurasi
1	2014jan-7 revisi - 4 hlm.docx	0,55	0,846	0,918
2	2013jul-6 revisi - 8 hlm.docx	0,885	1	0,986
3	Dampak Pembangunan Jembatan Suramadu...docx	0,695	1	0,96
4	Mental Imagery pada Korban Bullying.docx	0,75	1	0,996
5	Risk Analysis of Foregin and Joint Venture Banks.doc	0,523	1	0,958
	Rata-rata	0,6806	0,969	0,9636

B. Pembahasan

Dari pengujian yang dilakukan dapat diidentifikasi bahwa sistem masih mempunyai kekurangan dalam nilai *recall* yang merepresentasikan kemampuan sistem dalam mendapatkan informasi relevan yang didapatkan kembali dibandingkan jumlah informasi-informasi yang relevan. Pada kasus data uji 1 terdapat sebuah paragraf yang hanya

memiliki satu buah kalimat dan kalimat tersebut tidak diakhiri dengan tanda pengakhir kalimat seperti (.), (!) ataupun (?). Hingga pada akhirnya sistem tidak membacanya sebagai sebuah informasi kalimat yang valid. Dari kegagalan itu, pada akhirnya sistem tidak mampu mendeteksi tanda baca yang ada di dalam kalimat tersebut sehingga menimbulkan nilai *recall* yang kurang dari 1.

Hasil uji coba pada 5 data uji juga menunjukkan sistem masih mempunyai celah sehingga nilai presisi yang didapat tidak mendekati 1. Meskipun secara keseluruhan sistem masih dapat mendeteksi sebagian besar kesalahan tanda baca pada karya ilmiah tersebut. Namun berdasarkan perbandingan antara hasil pemeriksaan menggunakan sistem dan pemeriksaan secara manual terdapat banyak hasil dari kalimat yang disalahkan penggunaan tanda bacanya oleh sistem sedangkan pada pemeriksaan secara manual tidak diidentifikasi sebagai penggunaan tanda baca yang salah.

Setelah ditelaah lebih lanjut, persoalan ini terjadi karena pemeriksaan huruf kapital sesudah tanda baca menjadi tidak relevan jika terdapat istilah atau penggunaan kata yang mengharuskan penggunaan huruf kapital namun pada kasus kata biasa tidak menggunakan huruf kapital (penggunaan huruf kapital dapat digunakan pada kata khusus seperti nama orang, nama jalan atau nama perusahaan). Sedangkan di sisi lain, penggunaan algoritma Boyer-Moore dalam melakukan pemeriksaan tanda baca tidak dapat melakukan ekstraksi informasi yang terkandung dalam suatu kalimat secara eksplisit (misalnya apakah kata sesudah tanda baca merupakan nama orang, nama produk atau bukan). Pada kasus data uji 1 terdapat contoh kalimat “Dimulai dari COBIT 2.0, COBIT 3.0, COBIT 4.0, COBIT 4.1 dan terakhir yang baru dirilis pada pertengahan tahun 2012 adalah COBIT 5.0.” yang disalahkan penggunaan tanda bacanya oleh sistem. Jika suatu kalimat pada umumnya terdapat tanda baca koma (,), selanjutnya akan diikuti oleh karakter huruf kecil/bukan huruf kapital. Namun dikarenakan kata COBIT merupakan suatu singkatan produk, maka penggunaan huruf kapital diharuskan dalam penulisannya walaupun sebelumnya terdapat tanda baca koma (,). Kekurangan sistem ini akhirnya mengurangi nilai presisi yang didapatkannya. Kesalahan itu juga muncul beberapa kali pada data uji data uji lain yang dideteksi oleh sistem.

Pada data uji lain juga terdapat beberapa kasus pemeriksaan tanda baca yang dianggap salah oleh sistem. Contohnya pada data uji 5 terdapat kalimat “Traditional risk-return models in finance equate $V(X)$ with the expected value of option X and $R(X)$ with its variance”. Kalimat tersebut dianggap salah penggunaan tanda bacanya karena pada umumnya penggunaan tanda baca kurung buka (()) didahului oleh spasi. Namun pada data uji tersebut tanda baca kurung (()) digunakan dalam pendefinisian rumus sehingga penulisannya perlu digabung dengan karakter sebelumnya. Seharusnya masalah ini akan dapat diatasi ketika penulisan rumus matematika biasanya menggunakan fitur Equation pada aplikasi Microsoft Word. Sehingga nantinya ketika sistem melakukan pra-pemrosesan dokumen, karakter ini akan dihilangkan dan tidak diperiksa sebagai mana isi dari paragraf umumnya.

V. KESIMPULAN DAN SARAN

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut.

1. Sistem dapat membangkitkan telaah kesalahan tanda baca berdasarkan kesalahan yang dideteksi dari karya ilmiah serta penggunaan algoritma pencarian kata (dalam kasus ini menggunakan algoritma Boyer-Moore) dapat digunakan pada kasus-kasus pendeteksian kesalahan tanda baca yang berhubungan dengan penggunaan spasi atau tidak, serta penggunaan huruf kapital atau huruf normal.
2. Sistem dapat mendeteksi kesalahan tanda baca pada karya ilmiah berbahasa Indonesia menggunakan metode pencarian kata Boyer-Moore dengan rata-rata nilai presisi dan *recall* sebesar 0,6806 dan 0,969.
3. Sistem dapat melakukan deteksi dengan rata-rata tingkat akurasi sebesar 0,9636. Terdapat beberapa kesalahan tanda baca yang terdeteksi oleh sistem meskipun bukan merupakan penggunaan tanda baca yang salah.

Berdasarkan kesimpulan di atas pula, penulis memberikan saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Diantaranya adalah sebagai berikut.

1. Meningkatkan presisi dan *recall* sistem untuk mendeteksi adanya kesalahan tanda baca dengan menambahkan aturan-aturan pemeriksaan lain yang mungkin terjadi.
2. Melakukan pendekatan dengan menggunakan metode pengolahan bahasa alami untuk mendapatkan informasi yang lebih akurat dari karya ilmiah sehingga lingkup deteksi kesalahan tanda baca menjadi lebih luas dan tidak hanya mencakup pemeriksaan spasi dan penggunaan huruf kapital saja.

DAFTAR PUSTAKA

- [1] Dikti (2012). Surat Edaran Direktorat Jenderal Pendidikan Tinggi. Tersedia: <http://www.dikti.go.id/files/atur/SKDirjen152-E-T-2012KaryaIlmiah.pdf>.
- [2] Fakultas Teknik Universitas Muhammadiyah Sorong (2013). Jumlah Karya Tulis Indonesia Meningkat. Tersedia: <http://teknik.um-sorong.ac.id/berita-174--jumlah-karya-tulis-indonesia-meningkat.html>
- [3] Kemdikbud (2000). Pedoman Umum Ejaan Yang Disempurnakan. Tersedia: http://badanbahasa.kemdikbud.go.id/lamanbahasa/sites/default/files/pedoman_umum-ejaan_yang_disempurnakan.pdf.

- [4] C. Charra and T. Lecroq, Handbook of Exact String Matching Algorithms, King's College Publications, 2004.
- [5] Microsoft (2003). Primary Interop Assembly References. Tersedia: [http://msdn.microsoft.com/en-us/library/bb190883\(v=office.11\).aspx](http://msdn.microsoft.com/en-us/library/bb190883(v=office.11).aspx).
- [6] Cathal (2013). DocX. Tersedia: <http://docx.codeplex.com/>.
- [7] P. Raghavan, C. D. Manning and H. Schutze, "Evaluation in Information Retrieval," in Introduction to Information Retrieval, Cambridge University Press, 2008.